

增强云服务可信性的研究

王亚普¹, 王志坚^{1,2}, 叶 枫^{1,2}, 娄渊胜¹, 周发超¹

(1 河海大学 计算机与信息学院, 江苏 南京 211100;

2 南京航空航天大学 计算机科学与技术学院, 江苏 南京 210016)

摘 要: 结合自治愈机制与 AOP 技术, 提出了一种增强云服务的可信性框架. 该框架从云服务的逻辑实现层面来增加其可信性, 以“可插拔”的形式动态配置云服务的可信策略. 通过失效注入的方式进行仿真实验, 结果表明该框架有效地降低了云服务失效的概率, 增强了云服务的可信性.

关键词: 云服务; 可信性; 面向方面编程; 自治愈; 可插拔

中图分类号: TP311.5

文献标识码: A

文章编号: 1000-7180(2012)11-0038-04

The Research to Enhance the Dependability of Cloud Service

WANG Ya-pu¹, WANG Zhi-jian^{1,2}, YE Feng^{1,2}, LOU Yuan-sheng¹, ZHOU Fa-chao¹

(1 College of Computer and Information, Hohai University, Nanjing 211100, China;

2 College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China)

Abstract: This paper combines with the self-healing and AOP technology, and proposes a framework to enhance the dependability of cloud services. We consider enhancing the dependability from the logical layer of cloud service and implement “pluggable” way to configure the dependable strategies dynamically. Using fault injection method for simulation, the result shows that this framework can reduce the possibility of cloud service failure effectively and enhances the dependability of the cloud service.

Key words: cloud service; dependability; AOP; self-healing; pluggable

1 引言

由于云计算具有弹性、灵活性和开放性等特点, 而且云服务又是由不同的服务提供商设计、实现, 所涉及的可信性与安全性问题不容忽视.

要增强云服务的可信性, 日志监控、模型检测、数值分析、运行时监控、软件恢复、自治愈(self-healing)机制等均为可行方式^[1]. 其中, 自治愈机制源于自治计算, 它是指在系统在运行过程中能够自行判断何处存在问题, 并能实施相应处理的机制. 文献[2-5]介绍了当前自治愈机制在 Web 服务、软件可靠性、系统架构、分布式平台等方面的研究与应用,

然而以可插拔的方式、从逻辑层面考虑自治愈机制的设计与实现以及将自治愈机制与云服务相结合的研究与应用鲜有. 因此, 本文以增强云服务的可信性为切入点, 提出了一个云服务的可信性增强框架, 通过使用 AOP 将可信策略以“可插拔”的形式插入到云服务中, 最终实现增强云服务可信性的目标.

2 解决方案

2.1 框架的体系结构

图 1 是所提出的一种云服务的可信性框架的体系结构图, 该框架主要包括以下组件: ① 日志监控组件: 日志监控组件能在不改变云服务源代码的基

收稿日期: 2012-05-03; 修回日期: 2012-06-09

基金项目: 河海大学中央高校基本科研业务费项目(2009B21614); 河海大学自然科学基金项目(2009421211); 江苏省水利科技计划项目(2011024)

础上,以附加功能的形式获取相关日志记录;② 诊断模块:负责解析日志监控组件提供的错误报告,存储相关信息,并根据需要通知策略仓库设置解决问题的策略;③ 策略仓库:根据错误的分类以及历史记录为云服务查找合适的策略,根据 Java 的反射机制完成可信策略的映射关系以及实例化.该组件包括查询策略、添加策略、删除策略三个部分,在没有查询到合适的策略时,可以通过添加策略设置新的可信策略;④ 策略配置:为云服务生成可以插拔的可信策略;⑤ 自治愈合切面组:实现云服务在一定程度上自我修复的一组可信策略.⑥ 包装、合成模块:运行不可缺少的辅助工具,如编织器、语言转换工具等,负责云服务信息的包装以及云服务与有效的可信策略的合成.

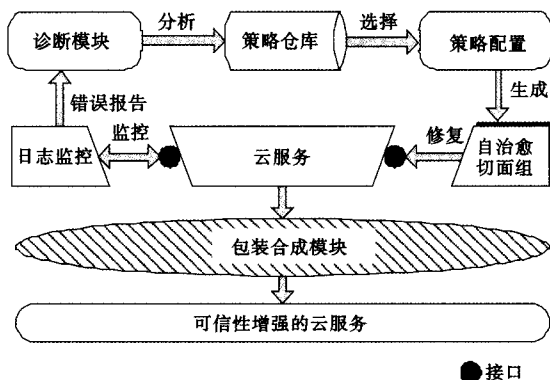


图1 云服务的可信性增强框架

上述组件之间的关系如下:通过日志监控切面对云服务的监控可以获取到用户信息、服务信息和错误报告,并将其提交给诊断组件,对错误进行归类、存档等操作,策略仓库根据分析结果选取合适的可信策略,生成相应可插拔的自治愈合切面组编织入云服务中.具体的实现流程如图2所示.

2.2 关键技术

2.2.1 日志监控

日志记录是一个安全、可靠的系统必须具备的功能,它可以作为配置云服务可信策略的重要依据.日志记录可以记录用户敏感操作、系统资源情况、虚拟机之间的调用等.本文使用 AOP^[6-7] 完成用户敏感操作记录和云服务运行时异常的监控,对云服务的监控主要是针对核心业务逻辑函数的监控.每当有用户登陆时就会启动一个新的线程,每个线程可以访问所授权的服务,通过使用 Agent 辅助日志监控切面来获得服务线程总数、每个用户所启用的服务、用户的操作记录、服务线程运行的起始和结束时

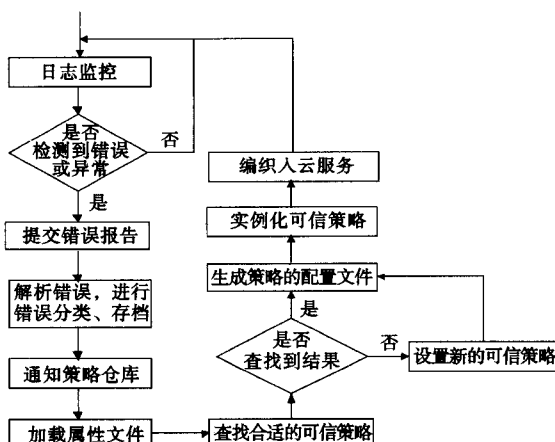


图2 框架实现的流程图

刻、服务线程的耗时、服务的运行状态、异常与错误信息等.

2.2.2 可信策略

在实际应用中,系统负载过大、客户端与服务器之间经常发生网络延迟、服务器宕机等情况均会导致服务消费者无法获得期待的结果,这时系统应该采取一些自治愈合的策略,降低云服务发生失效或错误的概率.本文利用时间冗余的原理完成以下两种自治愈合机制:① 重试(Retry)策略:主要用于解决瞬时性故障,当检测到云服务运行过程中出现故障时,重复执行故障指令,若故障是瞬时的,则在指令复执期间可能不会出现,服务就可以继续正常运行.② 回滚(Roll Back)策略:在云服务的逻辑层设置多个恢复点,在云服务发生错误的情况下可以从一个恢复点处开始重复执行一段程序.首先检验一小段程序的运行结果,若结果出现错误则进行回滚,再重复执行此部分程序,若一次回滚不能解决,可以多次回滚,直至故障消除.下面以重试策略作具体介绍,其核心代码描述如表1所示.

表1 重试策略的核心代码

```
Object around();Retry()
{try(Object result result=proceed();
}catch(RemoteException e)//任何远程方法调用实际上
要进行许多低级网络操作,因此网络错误可能在调用过
程中随时发生
{try
{Thread.sleep(1000);//线程休息1秒
result=proceed();//重试操作
}catch(InterruptedExcepion e1)
{e1.printStackTrace();}
}
return result;
}
```

2.2.3 “可插拔”机制

为了提高可信策略配置的灵活性和复用性,本文结合了工厂模式方法、Java 反射机制以及 AOP 的特性,通过属性文件来分别配置工厂类与具体可信策略的映射关系以及可信策略与云服务之间的映射关系,实现集配置与反射为一体的可插拔模型。可以根据需要增加或减少切面的数量,调整切面的顺序。

当有新的可信策略增加时,工厂方法模式可以有效地避免对工厂类代码的修改和重新编译的过程。当策略仓库在执行选择操作的时候,会自动加载工厂类与可信策略之间关系的属性文件,然后通过 Java 反射机制,系统便可以获得需要实例化的类。此外在云服务的逻辑层提供统一的访问接口,将可信策略独立成单独的模块编织入云服务之中,这样在切换不同可信策略的时候,只需要处理的是对工厂方法模式中实例化过程的切换。策略配置组件负责对不同云服务与工厂类之间映射关系的属性文件的解析,生成统一规格的可信策略的配置文件(XML 文件结构示例如表 2 所示)。然后通过实现云服务逻辑层的接口方法便可以方便地加载配置文件并对节点进行操作。

表 2 增加可信策略的文件结构

```
<? xml version="1.0" encoding="utf-8"? >
<addaspect>
  <cloudservice name="云服务 1" provider="wang">
    <ToS>SaaS</ToS>
    <classname>CloudManagerAction</classname>
    <methodname>service_custom</methodname>
    <parameterlist>
      <parameter>String</parameter>
    </parameterlist>
    <returntype>String</returntype>
    <exception>java.rmi.RemoteException</exception>
    <aspect>InstructionRetry</aspect>
  </cloudservice>
</addaspect>
```

以上述方式,就完成了可插拔的可信策略的创建。即便云服务运行环境发生变化而导致可信策略的需求改变,只需要再相应地修改配置文件,就能达到完全保留云服务逻辑层运行状态的目的。

3 仿真测试与结果分析

3.1 测试环境

实验中所使用到的云平台依托于项目“基于云计算的防汛防旱信息集成平台的研究”,建立在

Dell PowerEdge R900 服务器上。使用基于 Cloud-Stack 划分的装有 Windows Server 2008 操作系统的虚拟机作为 Web 服务器。其他软件包括 Tomcat v6.0, JDK v1.6.0, JMeter v2.6, 数据库采用 Mysql v5.1; 硬件包括 Xeon E7320, 4 组 4 核 2.13GHz CPU, 64G 内存。

3.2 模拟实验

为了验证可信策略的有效性,本文采用失效注入的方式来模拟错误的发生。失效注入的方式如下:在云服务程序中设置断点,当服务运行到断点位置就会产生一个软中断信号停止程序的执行,这时可以使用调试指令代替输入或输出指令来制造失效,比如抛出异常、修改变量等。为了便于模拟大批量用户访问的情况,采取如下更加简洁的失效注入操作:在待测的云服务中随机生成 0~10 的整数,若生成的随机整数在 6~7 范围之内,则注入异常或者错误;否则执行正常的操作。

本文使用 JMeter 模拟 10000 名用户并发访问待测云服务(云服务自身存在逻辑代码的错误以及负载压力过重的情况不作考虑),对比待测服务在有、无重试策略情况下的错误率,从而得出结果。为了保证测试结果的非偶然性,一共得到五组测试数据,对比结果如图 3 所示。

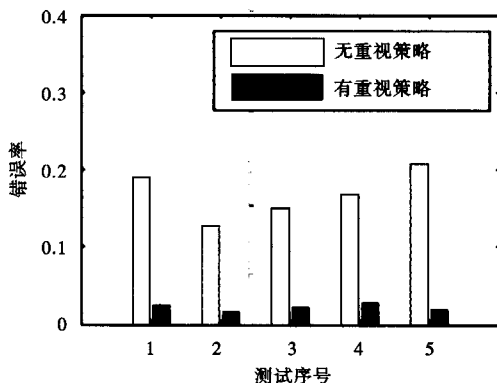


图 3 云服务在有、无重试策略情况下错误率对比

由图 3 可见,在客户端与服务器之间发生瞬时性通信错误时,可信框架所配置的重试策略大幅度降低了云服务发生失效或错误的概率,有效地增强了云服务的可信性。

4 结束语

本文以增强云服务的可信性为切入点,结合 AOP 和自愈的思想,提出了云服务的可信性增强框架。通过对错误报告的分析为云服务选择合理的

可信策略,以“可插拔”的形式编织到云服务中,降低云服务运行过程中发生失效与错误的概率,增强云服务的可信性.未来的研究工作主要集中在引入相关推理的算法,使得可信策略的选择更加智能、合理.

参考文献:

- [1] Avizienis A. Design of fault-tolerant computers[C]// Falut Joint Computer Conference. New York, 1967: 733-743.
- [2] Karray M, Ghedira C, Maamar Z. Towards a self-healing approach to sustain web services reliability[C]// Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on. Biopolis, 2011: 267-272.
- [3] Naccache H, Gannod G C. A self-healing framework for web services[C]// Web Services, 2007. ICWS 2007. IEEE International Conference on. Salt Lake City, 2007: 398-345.
- [4] Frincu M E. Self-healing distributed scheduling platform[C]// Cluster, Cloud and Grid Computing (CC-Grid), 2011 11th IEEE/ACM International Symposium on. Newport Beach, 2011: 225-234.
- [5] Alessandra Gorla, Mauro Pezzé, Jochen Wuttke, et al. Achieving cost-effective software reliability through self-healing[J]. Computing and Informatics, 2010, 29 (1): 93-115.
- [6] Vladimir O Safonov. Using AOP for trustworthy software development[M]. Canada: Wiley-Blackwell, 2008: 10-49.
- [7] Carsten Köllmann, Lea Kutvonen, Peter Linington, et al. An aspect-oriented approach to manage QoS dependability dimensions in model driven development [C]//International Workshop on Model-Driven Enterprise Information Systems. Portugal, 2007: 85-94.

作者简介:



王亚普 男, (1990—), 硕士研究生. 研究方向为云计算.

王志坚 男, (1958—), 教授, 博士生导师. 研究方向为软件理论.

叶枫 男, (1980—), 讲师, 博士研究生. 研究方向为云计算、水信息学.